

WHAT IS CLAIMED IS:

1. An adaptive accessing system for single level strongly consistent cache, comprising:

a server having a cache, at least one cached data entry, and a first counter and a second counter corresponding to each client of each cached data entry, the first counter measuring the number of cycles in an observed period, the second counter measuring the number of cycles that have updates in the cycles, wherein a cycle is defined as a period between two consecutive data accesses;

at least one client connected to the server via a communication link, each client having a cache; and

a dynamic adjustment module corresponding to each client of each cached data entry for selecting a poll-each-read algorithm or a callback algorithm based on a ratio of the first counter and the second counter to maintain a consistency of the caches in the client and the server.

2. The system as claimed in claim 1, wherein the dynamic adjustment module selects the poll-each-read algorithm if the ratio of the first and the second counters is greater than $1/2$, otherwise selects the callback algorithm.

3. The system as claimed in claim 1, wherein the first counter is incremented when the poll-each-read algorithm is selected and the server receives a cached data entry access request from the client.

4. The system as claimed in claim 3, wherein, when the client desires to access a cached data entry existed in the cache thereof, and the server has received the cached data entry access request from the client

and the cached data entry is invalid, the second counter is incremented.

5 5. The system as claimed in claim 1, wherein each cached data entry in the client has a third counter for measuring the number of accesses since a previous update, and when the callback algorithm is used and the client accesses the cached data entry in the cache thereof, the third counter is incremented.

6. The system as claimed in claim 5, wherein when the server updates the cached data entry thereof, the second counter is incremented.

10 7. The system as claimed in claim 6, wherein if a cached data entry in the client is set to be invalid, the client sends a value of the third counter to the server and sets the value of the third counter to be zero, and the server adds the value of the third counter to the first counter.

15 8. The system as claimed in claim 1, wherein when the value of the first counter is greater than a predetermined value, the server selects the poll-each-read algorithm or the callback algorithm by a ratio of the first counter and the second counter, and then sets both the first and the second counters to be zero.

9. The system as claimed in claim 1, wherein the communication link is wired link.

20 10. The system as claimed in claim 1, wherein the communication link is a wireless link.

25 11. An adaptive accessing method for single level strongly consistent cache, capable of selecting a poll-each-read algorithm or a callback algorithm to maintain a consistency of caches between a server and at least one client, the method comprising the steps of:

(A) in the server, using a first counter for measuring the number of cycles in an observed period, and a second counter for measuring the number of cycles that have updates in the cycles, wherein a cycle is defined as a period between two consecutive data accesses;

5 (B) determining a ratio of the first counter and the second counter;
and

(C) selecting a poll-each-read algorithm or a callback algorithm based on the ratio.

10 12. The method as claimed in claim 11, wherein in step (C), the poll-each-read algorithm is selected if the ratio is greater than 1/2; otherwise the callback algorithm is selected.

13. The method as claimed in claim 11, wherein in step (A), the first counter is incremented when the poll-each-read algorithm is selected and the server receives a cached data entry access request from the client.

15 14. The method as claimed in claim 13, wherein when the client desires to access a cached data entry existed in the cache thereof, and the server has received the cached data entry access request from the client and the cached data entry is invalid, the second counter is incremented.

20 15. The method as claimed in claim 11, wherein in the step (A), each cached data entry in the client has a third counter for measuring the number of accesses since a previous update, and when the callback algorithm is used and the client accesses the cached data entry in the cache thereof, the third counter is incremented.

25 16. The method as claimed in claim 15, wherein when the server updates the cached data entry thereof, the second counter is incremented.

17. The method as claimed in claim 16, wherein if a cached data entry in the client is set to be invalid, the client sends a value of the third counter to the server and sets the value of the third counter to be zero, and the server adds the value of the third counter to the first counter.

5 18. The method as claimed in claim 11, wherein after executing the step (C), both the first and the second counters are set to zero.

20200922 10:57:27